บุญเลิศ เอี่ยมทัศนา

TPW
: จับภาพใส่หน้าต่าง

ศึกษาเทคนิค วิธีการ จับภาพ ลอกภาพ การขยายและย่อภาพ และการนำภาพออกแสดง

ต้นเรื่องของโปรแกรมนี้เป็นการผูกเรื่องเพื่อจะกล่าวถึงการลอกภาพบิตแม็พด้วยวิธีและลักษณะต่าง ๆ กัน และแสดงการติดต่อผ่านทางคลิปบอร์ด กับที่มาเพิ่มในภายหลัง คือ การใช้ไดอะล็อกบ็อกซ์กลางของวินโดวส์ ตามที่กล่าวไว้ในตอนที่แล้ว

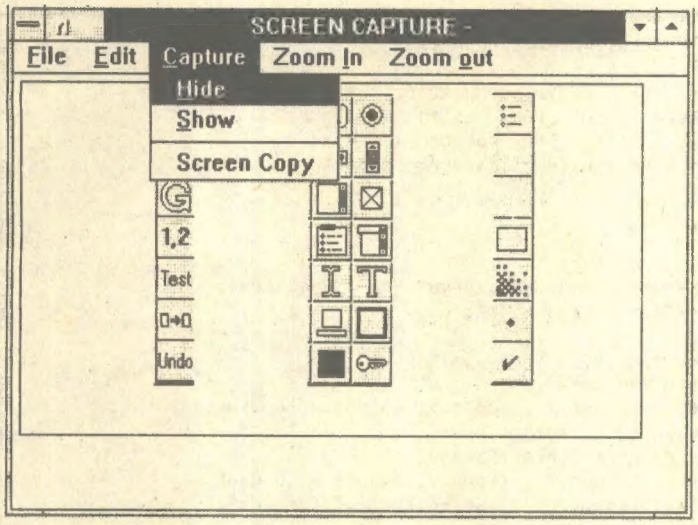
แต่ตัวของมันเองก็เป็นโปรแกรมที่ใช้งานได้ดีพอสมควร เสริมช่องว่างที่เพนต์บรัช (Paint Brush) ทำไม่ได้ คือ ใช้จับภาพบนจอภาพซึ่งอาจจะลอกมาโดยตรง หรือลอกในแบบย่อขยาย หรือลอกแล้วนำมาขยายเพื่อตัดภาพขนาดเล็ก (เพนต์บรัชขยายแล้วตัดภาพไม่ได้) หรือจับภาพทั้งจอภาพ (ซึ่งใช้เพนต์บรัชไม่ได้อีกเช่นกัน)

ตอนต่อ

โปรแกรมในฉบับนี้เป็นตอนต่อของฉบับที่แล้ว (ไมโครฯ 94) เป็นโปรแกรมที่จะต้องป้อนแทรกป้อนต่อกับโปรแกรม ScrnCaptor2a เป็น 2b

ลิสติงที่ 1.1

```
PROGRAM  ScrnCaptor2b;

TYPE

  CaptModeT = (nocapt, hhde, sshw, ccpy, mmrk, mrkok);

  WndProCP =  WndProC;
  WndProC = OBJECT(TWindow)
    ..............
    cptmde : CaptModeT;
  procedure  cmCaptHide(var msg : TMessage);
              virtual cm_First + cm_CaptHide;
  procedure  cmCaptShow(var msg : TMessage);
              virtual cm_First + cm_CaptShow;
  procedure  cmScrCopy(var msg : TMessage);
              virtual cm_First + cm_ScrCopy;
  procedure  cmZoomIn(var msg : TMessage);
              virtual cm_First + cm_ZoomIn;
  procedure  cmZoomOut(var msg : TMessage);
              virtual cm_First + cm_ZoomOut;
  procedure  wmLButtonDown(var msg : TMessage);
              virtual wm_First + wm_LButtonDown;
  procedure  wmMouseMove(var msg : TMessage);
              virtual wm_First + wm_MouseMove;
  procedure  wmLButtonUp(var msg : TMessage);
              virtual wm_First + wm_LButtonUp;
  procedure  cmMark(var msg : TMessage);
              virtual cm_First + cm_Mark;
  procedure  cmGetMark(var msg : TMessage);
              virtual cm_First + cm_GetMark;
  procedure  cmGetMarkBMP(var msg : TMessage);
              virtual cm_First + cm_GetMarkBMP;
  procedure  ScrnToBMP(var dstnBMP : HBitmap;
                       var bmpwid,bmphi : LongInt);
  procedure  BMPtoBMP(var dstnBMP : HBitmap; dx,dy : Integer;
                      srceBMP : HBitmap, sx,sy,swid,shi : Integer
  procedure  DoCopyToWnd;
  procedure  DoMarkOff;
    ..................
             END;


(* ------------------------------------------------ *)


procedure  SwapPoint(var frst,scnd : Integer);
var   tem : Integer;
begin
  if  frst > scnd  then
    begin  tem := frst;  frst := scnd;  scnd := tem  end;
end;
```



รูปที่ 1
เมนูของโปรแกรม
ScrnCaptor

ดังการกำหนดกิจกรรมเพิ่มในลิสติงที่ 1.1 แต่ได้นำการกำหนดแบบข้อมูล CaptModeT และฟิลด์ข้อมูล cptmde มาแสดงซ้ำเพื่อที่จะได้ไม่ต้องพลิกไปอ่านในตอนที่แล้ว

แนะนำเมนู

ขอเริ่มที่เมนูตามรูปที่ 1 (ซึ่งผมได้จับภาพเครื่องมือของไดอะล็อกเอดิเตอร์รีซอร์สเวอร์กชอปมาแสดงเพื่อไม่ให้หน้าต่างว่างเท่านั้น) มีรายการที่จะขอนำมากล่าวถึงก่อน คือ

1. รายการ Capture มีรายการรองให้เลือกอีกดังนี้ คือ Hide, Show และ Screen Copy ซึ่งจะมีการทำงานเป็นส่วนรวมดังนี้ คือ เมื่อเลือกรายการเหล่านี้ เคอร์เซอร์จะเปลี่ยนเป็นรูปกากบาท ให้คลิ๊กเม้าส์ซ้ายที่มุมบนขวาของภาพที่ต้องการจับ เลื่อนเคอร์เซอร์ไปที่มุมล่างขวาแล้วปล่อยมือ เคอร์เซอร์จะเปลี่ยนกลับเป็นรูปลูกศรตามเดิม จะเป็นการจับภาพตามลักษณะต่าง ๆ กันดังนี้

- Screen Copy จะเป็นการลอกภาพจากจอภาพไปให้แก่หน้าต่างโดยตรง ไม่ได้เก็บภาพไว้ ดังนั้น เมื่อผู้ใช้เปลี่ยนขนาดของหน้าต่าง ภาพในหน้าต่างจะถูกลบ

- Hide และ Show เป็นการเก็บภาพ คือ ได้ให้เก็บภาพไว้ในแผ่นภาพบิตแม็พก่อนแล้วจึงนำออกแสดง โดยที่ Hide จะเก็บภาพในแบบหายตัวคือ ไม่แสดงหน้าต่างในขณะเก็บภาพ ทำให้สามารถเก็บภาพขนาดใหญ่ได้ไม่เกะกะกัน ส่วน Show จะแสดงตัวตลอดเวลา จุดประสงค์คือ เพื่อใช้เก็บภาพในหน้าต่างของตนเองอีกทอดหนึ่ง

2. รายการ Edit มีรายการเกี่ยวกับการอิดิตหลายรายการ ที่เกี่ยวกับตอนนี้คือ Mark เป็นการหมายพื้นที่ด้วยการตีกรอบรอบภาพในหน้าต่าง โดยเลือกรายการนี้แล้วคลิ๊กที่มุมบนซ้ายเลื่อนเคอร์เซอร์ (ซึ่งจะทำให้เกิดการตีกรอบตามไปด้วย) ไปยังมุมล่างขวาแล้วปล่อยมือ ทั้งนี้เพื่อการลอก (copy) หรือตัด (cut) กับคลิปบอร์ด แต่ในโปรแกรมนี้ใช้งานอื่นด้วย

กิจกรรมจับภาพ

ได้จัดกิจกรรมตอบรับเมสเสจจากเมนูดังกล่าวมาตามลำดับ และมีชื่อที่ใกล้เคียงกัน ดังนี้คือ cmCaptHide, cmCaptShow, cmScrCopy และ cmMark ตามโปรแกรมในลิสติงที่ 1.2 มีการทำงานที่เหมือนกันดังนี้คือ

```
PROCEDURE  WndProC.cmCaptHide(var msg : TMessage);
BEGIN
  SetCapture(HWindow);   cptmde := hhde;
  SetCursor(LoadCursor(0, idc_Cross));
  MoveWindow(HWindow,640,480,0,0,false)
END;


PROCEDURE  WndProC.cmCaptShow(var msg : TMessage);
BEGIN
  SetCapture(HWindow);   cptmde := sshw;
  SetCursor(LoadCursor(0, idc_Cross));
END;


PROCEDURE  WndProC.cmScrCopy(var msg : TMessage);
BEGIN
  SetCapture(HWindow);   cptmde := ccpy;
  SetCursor(LoadCursor(0, idc_Cross));
END;


PROCEDURE  WndProC.cmMark(var msg : TMessage);
BEGIN
  SetCapture(HWindow);   cptmde := mmrk;
  SetCursor(LoadCursor(0, idc_Cross));
  if  mrkon  then  DoMarkOff;
END;
```

ลิสติงที่ 1.2

- ให้โหมดของตนแก่ cptmde ทั้งนี้เพราะการแจกแจงที่ทำให้เกิดเป็นการจับภาพลักษณะต่าง ๆ จะต้องอยู่ในโปรแกรมของกิจกรรม wmLButtonUp

- SetCapture เพื่อให้สามารถใช้งานเม้าส์นอกหน้าต่างของตนได้

- เปลี่ยนภาพเคอร์เซอร์ใหม่เป็นภาพกากบาท

สำหรับการทำงานที่แตกต่างกันคือ

- ใน cmCaptHide ใช้ MoveWindow ย้ายหน้าต่างออกไปไว้นอกจอภาพเพื่อซ่อนหน้าต่าง ซึ่งอาจใช้ฟังก์ชัน ShowWindow ซ่อนหน้าต่างได้เช่นกัน แต่จะซ่อนต่อสายตาผู้ใช้เท่านั้น เมื่อให้จับภาพจะจับภาพของตนเองได้

- ใน cmMark ได้ให้ตรวจว่าถ้ามีการหมายพื้นที่อยู่เดิมให้ยกเลิกด้วย DoMarkOff

กิจกรรมสนองตอบเม้าส์

ดังที่ผมเคยกล่าวแล้วว่า การเขียนโปรแกรมบนวินโดวส์เป็นการเขียนโปรแกรมเป็นท่อน ๆ ไม่ต่อเนื่องกัน เพราะเป็นการเขียนโปรแกรมตอบสนองเหตุที่เกิดจากผู้ใช้ ดังเช่นโปรแกรมนี้เมื่อให้ผู้ใช้เลือกรายการก็เขียนโปรแกรมตอบสนองการเลือกรายการและก็ทิ้งไว้เท่านั้น

เมื่อต้องการให้ผู้ใช้ใช้เม้าส์ในการจับภาพ ก็จะต้องมาเขียนโปรแกรมท่อนต่อไปเพื่อตอบรับการใช้เม้าส์ของผู้ใช้ ดังโปรแกรมในลิสติงที่ 1.3 ได้แก่

```
PROCEDURE  WndProC.wmLButtonDown(var msg : TMessage);
BEGIN
  if  mrkon  then  begin  DoMarkOff;  Exit  end;
  if  cptmde = nocapt  then  Exit;
  if  cptmde = mmrk  then  cptmde := mrkok;
  sttpnt := MakePoint(msg.lParam);  endpnt := sttpnt;
END;


PROCEDURE  WndProC.wmMouseMove(var msg : TMessage);
VAR  rct : TRect;   aDC : HDC;
BEGIN
  if  cptmde <> mrkok  then  Exit;
  adc := GetDC(HWindow);
  SetRect(rct, sttpnt.x ,sttpnt.y, endpnt.x, endpnt.y);
  DrawFocusRect(adc, rct);
  endpnt := MakePoint(msg.lParam);
  SetRect(rct, sttpnt.x ,sttpnt.y, endpnt.x, endpnt.y);
  DrawFocusRect(adc, rct);  ReleaseDC(HWindow,adc);
END;


PROCEDURE  WndProC.wmLButtonUp(var msg : TMessage);
VAR   wndDC,memDC : HDC;  hOldObj : THandle;
      arect : TRect;  res : Bool;  temBmp : HBitmap;
BEGIN
  if  cptmde = nocapt  then  Exit;
  ReleaseCapture;
  endpnt := MakePoint(msg.lParam);

  if  (cptmde = hhde) or (cptmde = sshw)  then
    begin  ScrnToBMP(mainBMP,mainwid,mainhi);  SetDefault  end;

  if  cptmde = hhde  then  MoveWindow(HWindow,0,0,32,56,false)
  else if  cptmde = sshw  then  InvalidateRect(HWindow, nil, true)
  else if  cptmde = ccpy  then  DoCopyToWnd
  else if  cptmde = mrkok  then
    begin  cptmde := mmrk;  mrkon := true  end;
  cptmde := nocapt;
END;


PROCEDURE  WndProC.DoMarkOff;
VAR  rct : TRect;   aDC : HDC;
BEGIN
  adc := GetDC(HWindow);  mrkon := false;
  SetRect(rct, sttpnt.x ,sttpnt.y, endpnt.x, endpnt.y);
  DrawFocusRect(adc, rct);  ReleaseDC(HWindow,adc);
END;
```

ลิสติงที่ 1.3

1. wmLButtonDown

สนองตอบเมสเสจอันเกิดจากการที่ผู้ใช้กดปุ่มซ้ายของเม้าส์ในเที่ยวขาลง คือ เมื่อกดลงได้ให้ดำเนินการดังนี้คือ

- ถ้าขณะนั้นมีการหมายพื้นที่อยู่ คือ mrkon เป็นจริงให้ยกเลิกคือลบออกเพื่อให้สามารถยกเลิกการหมายพื้นที่ได้ด้วยการคลิ๊กเม้าส์

- ถ้าโหมดเป็น nocapt คือ ปกติไม่ได้มีการเลือกรายการใด ไม่ต้องดำเนินการต่อ

- ถ้าโหมดเป็น mmrk คือการหมายพื้นที่ให้เปลี่ยนโหมดเป็น mrkok เพื่อแสดงว่าผู้ใช้ได้คลิ๊กเม้าส์แล้ว

- เก็บจุดเริ่มต้น ซึ่งเป็นตำบลที่ของเม้าส์เมื่อกดใน msg.lParam ไว้ใน sltpnt

2. wmMouseMove

วินโดวส์จะส่งเมสเสจนี้เมื่อผู้ใช้เลื่อนเม้าส์ ไม่ว่าจะกดปุ่มหรือไม่ จึงได้ให้เปลี่ยนโหมดเป็น mrkok ตามที่

การใช้ Screen Captor

หมายภาพ

เมื่อเลือกรายการที่ให้หมายภาพ เคอร์เซอร์จะเปลี่ยนเป็นรูปกากบาท ให้

- คลิ๊กซ้ายที่มุมบนซ้ายของภาพ
- เลื่อนเคอร์เซอร์ไปยังมุมล่างขวา แล้วปล่อยมือ ถ้าเป็นรายการ Edit/Mark จะแสดงกรอบสี่เหลี่ยมเส้นประตามไปด้วย

จับภาพ

ทั้งจอภาพ ให้

- กดคีย์ Print Screen ซึ่งอาจกดก่อนรันโปรแกรมนี้
- เลือกรายการ File/Get/Clipboard

จับภาพโดยตรง รายการต่อไปนี้ใช้สำหรับจับหรือลอกภาพในลักษณะต่าง ๆ กันคือ

- Capture/Hide ใช้จับภาพขนาดใหญ่ในขณะที่ซ่อนคือ ไม่แสดงหน้าต่างนี้บนจอภาพ
- Capture/Show ใช้ในการจับภาพขนาดเล็กนอกหน้าต่าง หรือจับ (คือตัด) ภาพที่จับมาแล้ว
- Capture/Screen Copy เป็นการลอกภาพมาใส่หน้าต่าง โดยที่ถ้าหน้าต่างเล็กกว่าภาพจะเป็นการย่อ หากใหญ่กว่าจะเป็นการขยาย แต่จะไม่เก็บภาพไว้ เช่น เมื่อผู้ใช้เปลี่ยนขนาดของหน้าต่างภาพที่ลอกมาจะหายไป

เมื่อเลือกรายการเหล่านี้แล้วให้หมายภาพ จะได้ภาพในหน้าต่างและอยู่ในกรอบของแผ่นภาพ สำหรับ Hide และ Show ขนาดของแผ่นภาพจะขึ้นกับขนาดของภาพไม่ขึ้นกับหน้าต่าง

ตัดภาพ

สามารถตัดภาพที่จับมาหรือโหลดจากไฟล์ หรือปะจากคลิปบอร์ด ได้ดังนี้

- อาจขยายภาพด้วย Zoom In (และย่อกลับด้วย Zoom Out)
- เลือกรายการ Edit/Mark เพื่อหมายภาพ
- เลือกรายการ Get/Mark จะเป็นการตัดภาพจากจอภาพ

หรือ ถ้าเป็นภาพที่ขยายไว้หากต้องการตัดเท่าของจริง (ก่อนการขยาย) ให้เลือกรายการ File/Get/BMP Mark (อาจใช้ Capture/Show ตัดภาพแทนได้)

การลอกและตัดลงคลิปบอร์ด

- เลือกรายการ Edit/Mark เพื่อหมายภาพ
- เลือกรายการ Edit/Copy เพื่อลอก หรือ Edit/Cut เพื่อตัด (แต่จะตัดภาพที่ขยายไว้ไม่ได้)

การปะจากคลิปบอร์ด

- อาจต้องเตรียมแผ่นภาพใหม่ โดยใช้ Capture/Show ทำเป็นแผ่นภาพว่าง
- เลือกรายการ Edit/Paste จะแสดงภาพจากคลิปบอร์ดที่มุมบนซ้ายของแผ่นภาพ
- หรือเลือกรายการ Edit/Mark แล้วคลิ๊กเม้าส์ ณ ตำแหน่งที่ต้องการให้เป็นจุดมุมบนซ้ายของการปะ แล้วเลือกรายการ Edit/Paste at Mark
- หากต้องการยกเลิกให้คลิ๊กเม้าส์ขวา
- เมื่อได้ตำแหน่งที่ต้องการแล้วให้เลือกรายการ Edit/Paste OK เพื่อปะลงแผ่นภาพ

อีกวิธีหนึ่งคือ เลือกรายการ File/Get/Clipboard ซึ่งจะได้แผ่นภาพเท่ากับภาพในคลิปบอร์ด ดังที่ใช้จับภาพทั้งจอภาพ

ไฟล์

ตามการใช้งานไฟล์ของวินโดวส์ รายการที่ไม่มีโปรแกรมคือ File/New, File/Printer setup, Edit/Clear, Edit/Copy to และ Edit/Paste fram

กล่าวมา เพื่อแสดงว่าผู้ใช้ได้เลือกรายการ Edit/Mark และกดปุ่มซ้ายของเม้าส์แล้วจึงจะดำเนินการต่อไปคือ

ให้ตีกรอบสี่เหลี่ยมเส้นประด้วย DrawFocusRect ซึ่งจะเป็นการตีกรอบด้วยวิธี XOR คือ หากให้ตีซ้ำจะเป็นการลบ จึงให้ตี 2 ครั้งโดยที่ครั้งแรกจะเป็นการลบของเดิม และครั้งที่สองจะแสดงกรอบค้างอยู่

ทั้งนี้เพราะครั้งแรกแท้ ๆ wmLButtonDown จะให้จุดเริ่มต้นคือ sltpmt กับจุดหลังคือ endpmt ตรงกัน ทำให้ DrawFocusRect ไม่ตีกรอบเท่ากับเป็นการลบเช่นกัน

3. wmLButtonUp

เมื่อผู้ใช้ปล่อยมือจากเม้าส์จะมาทำงานที่กิจกรรมนี้ เช่นกัน ได้ให้ตรวจว่าต้องเป็นการใช้เม้าส์หลังจากเลือกรายการดังกล่าวมาจึงจะให้ดำเนินการต่อไปคือ

- ยกเลิกการครอบครองเม้าส์ด้วย ReleaseCapture
- เก็บจุดหลังไว้ใน endpnt
- หากเป็นการเก็บภาพไม่ว่าจะเป็น Hide หรือ Show ให้เก็บภาพที่มีจุดมุมบนซ้ายอยู่ที่ sltpnt.x กับ

sltpnt.y และมีจุดมุมล่างขวาอยู่ที่ endpnt.x กับ endpnt.y ไว้ใน mainBMP (ตามที่กล่าวในตอนที่แล้ว) ด้วยกิจกรรม ScrnToBMP (ตามที่จะกล่าวต่อไป) ซึ่งจะให้ค่าความกว้างและความสูงของภาพคืนมาด้วยตามที่ให้เก็บไว้ใน mainwid และ mainhi

สำหรับการดำเนินการต่อไป ได้ตรวจค่าโหมดใน cptmde ซึ่งถ้าเป็น

- hhde คือ Hide ให้ย้ายหน้าต่างออกมาแสดง ซึ่งจะทำให้เกิดการทำงานที่ Paint (ตามที่กล่าวในตอนที่แล้ว) นำภาพใน mainBMP ออกแสดง

- sshw คือ Show ให้ InvalidateRect บังคับไปทำงานที่ Paint เพื่อนำภาพใน mainBMP ออกแสดงเช่นกัน

- ccpy คือ Screen Copy ให้ DocopyToWnd ตามที่จะกล่าวต่อไป

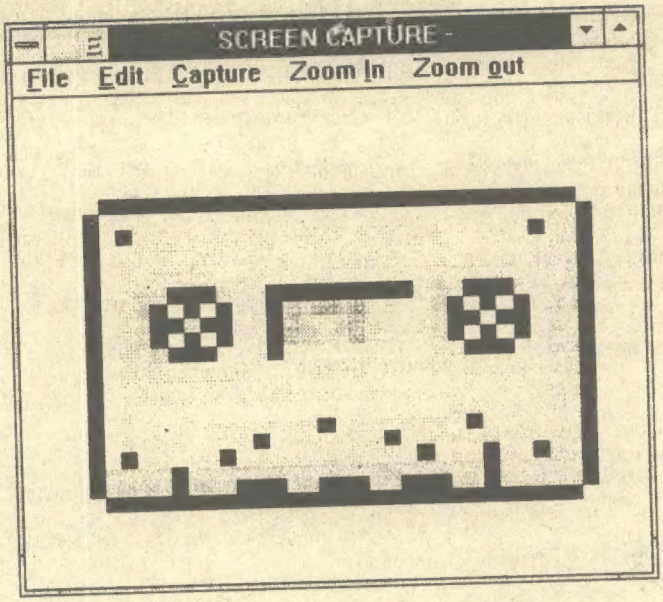
- mrkok คือ Mark ให้สถานะ mrkon เป็นจริงเพื่อแสดงว่าได้มีการหมายพื้นที่เพื่อการอิดิตไว้

ท้ายสุดได้ให้ค่าสถานะ nocapt แก่ cptmde เพื่อเป็นการยุติการทำงานตามการเลือกรายการดังกล่าวมา

โปรแกรมยามว่าง : การลอกภาพ

โปรแกรมตามลิสติงที่ 1.4 เป็นโปรแกรมที่ผมเขียนขึ้นมาก่อน ยามว่างก็ใช้จับภาพต่าง ๆ ส่วนมากเป็นไอคอนมาขยายดูเล่นครับ ดังรูปที่ 2 เป็นภาพไอคอนหนึ่งในโปรแกรมแมนเนเจอร์ โดยใช้ฟังก์ชัน StretchBlt อีกสองฟังก์ชันที่เกี่ยวกับการลอกภาพคือ BitBlt และ PatBlt

โดยที่คำว่า Blt (อ่านว่า blit) ย่อมาจาก block transfer หมายถึง การย้ายข้อมูลเป็นกลุ่มก้อน คือเป็นพื้นที่



รูปที่ 2
แสดงการจับภาพมาขยายด้วยฟังก์ชัน StretchBlt

```
PROCEDURE  WndProG.DoCopyToWnd;
VAR   adc : HDC;  arect : TRect;
BEGIN
  adc := GetDC(HWindow);
  GetClientRect(HWindow, arect);
  StretchBlt(adc,0,0, arect.right, arect.bottom,
             adc, sttpnt.x, sttpnt.y,
             endpnt.x-sttpnt.x, endpnt.y-sttpnt.y, SrcCopy);
  ReleaseDC(HWindow, adc);
END;
```

ลิสติงที่ 1.4

และสำหรับ StretchBlt จะเป็นการย้ายในแบบย่อหรือขยาย คือถ้าตำบลที่ปลายทางเล็กกว่าต้นทางจะเป็นการย่อ ถ้าใหญ่กว่าจะเป็นการขยาย

สำหรับโปรแกรมของกิจกรรม DoCopyToWnd ในลิสติงที่ 1.4 ตามที่ wmLButtonUp จะเรียกใช้เมื่อผู้ใช้เลือกรายการ Screen Copy เป็นการใช้งานฟังก์ชัน StretchBlt อย่างพื้นฐานที่สุด เป็นการลอกภาพโดยตรงจากจอภาพให้แก่จอภาพ โดยที่ StretchBlit ต้องการค่าพารามิเตอร์ 3 กลุ่มคือ

1. กลุ่มแรกประกอบด้วย ดีซี (DC : device context) ตำแหน่ง (จุดมุมบนซ้าย) และขนาด (ความกว้างและความสูง) ของพื้นที่ปลายทาง ค่าที่ให้คือ

- ค่าดีซีของหน้าต่าง ซึ่งถ้าเป็นค่าดีซีของเครื่องพิมพ์ก็จะเป็นการลอกภาพออกทางเครื่องพิมพ์

- ค่าตำแหน่งคือ 0,0 กับ arect.right และ arect.buttom ซึ่งเป็นค่าความกว้างและความสูงของพื้นที่ใช้งาน (Client area) คือเต็มหน้าต่าง ซึ่งชื่อของฟิลด์ใน TRect อาจทำให้งงอยู่บ้าง แต่ความจริงแล้วค่าที่ได้เมื่อใช้ GetClientRect จะเป็นค่าดังที่นำมาใช้

2. กลุ่มที่สองประกอบด้วย ดีซี ตำแหน่งและขนาดเช่นกัน แต่เป็นของพื้นที่ต้นทาง ค่าที่ให้คือ

- ค่าดีซีของหน้าต่าง นั่นคือ เทียบได้กับเป็นการย้ายภาพในหน้าต่างเดียวกัน ถึงแม้พื้นที่ต้นทางจะออกไปอยู่นอกหน้าต่าง

```
PROCEDURE  WndProC.ScrnToBMP(var dstnBMP : HBitmap;
                             var bmpwid,bmphi : LongInt);
VAR   wndDC,dstnDC : HDC;  hOldObj : THandle;
      arect : TRect;  res : Bool;
BEGIN
  if  dstnBMP <> 0  then  DeleteObject(dstnBMP);
  SwapPoint(sttpnt.x, endpnt.x);  SwapPoint(sttpnt.y, endpnt.y);
  bmpwid := endpnt.x - sttpnt.x;  bmphi  := endpnt.y - sttpnt.y;

  wndDC := GetDC(HWindow);
  dstnBMP := CreateCompatibleBitmap(wndDC, bmpwid, bmphi);
  if  dstnBMP = 0  then
    begin
    MessageBox(HWindow,'Can not create bitmap', 'Error', mb_ok);
    end;

  dstnDC := CreateCompatibleDC(wndDC);
  hOldObj := SelectObject(dstnDC, dstnBMP);
  res := BitBlt(dstnDC, 0, 0, bmpwid, bmphi,
                wndDC, sttpnt.x, sttpnt.y, SrcCopy);

  ReleaseDC(HWindow, wndDC);
  SelectObject(dstnDC, hOldObj);  DeleteDC(dstnDC);
END;
```

ลิสติงที่ 1.5

- ค่าตำแหน่งและขนาดของพื้นที่ต้นทางตามที่เก็บไว้ใน sltpnt กับ endpnt

3. กลุ่มที่สามมีค่าเดียว เป็นค่า rop คือ raster operation เพื่อบอกว่าต้องการให้ดำเนินการอย่างไร ค่า SrcCopy หมายถึงให้เป็นการลอก หากให้เป็นค่าอื่นเช่น SrcInvert จะเป็นการลอกแบบกลับสี เช่นขาวเป็นดำ

ก็ต้องทดลองใช้ดูครับจะปรากฏภาพในหน้าต่างทันทีที่ถอนมือจากเม้าส์ และสำหรับฟังก์ชันนี้การกำหนดพื้นที่ต้นทาง ไม่จำเป็นต้องเริ่มจากมุมบนซ้าย หากเริ่มจากมุมอื่นจะได้ภาพกลับลักษณะต่าง ๆ กัน เป็นโปรแกรมยามว่าง คือ เมื่อยังคิดอะไรไม่ออกได้เป็นอย่างดีครับ

การเก็บภาพ

เมื่อผู้ใช้เลือกรายการ Capture/Hide หรือ Show ในกิจกรรม wmLButtomUp จะเรียกใช้ ScrnToBMP ตามลิสติงที่ 1.5 เพื่อเก็บภาพไว้ใน mainBMP ตามที่ผ่านมาให้แก่ dstnBMP ซึ่งได้ให้ดำเนินการดังนี้

- ให้ยกเลิกแผ่นภาพเดิม (หากมี)

- สลับค่าใน sttpnt กับ endpnt ให้เป็นการเริ่มจากมุมบนซ้ายเสมอ เพราะการลอกภาพด้วย BitBlt ไม่มีการกลับภาพ

- สร้างแผ่นภาพเปล่า ฟังก์ชัน CreateCompatibleBitmap จะให้กำเนิดแผ่นภาพเปล่าตามขนาดที่ผ่านไปให้คือ bmpwid และ bmphi เป็นความกว้างและความสูงของพื้นที่ต้นทาง การให้กำณิดนี้หมายถึง การจัดทำส่วนหัวของแผ่นภาพ ซึ่งมีรายละเอียดอยู่หลายเรื่อง และการจองหน่วยความจำที่จะเก็บภาพด้วย และเช่นกัน DeleteObject จะยกเลิกหน่วยความจำนี้ (หากต้องการดูเรื่องเทียบเคียงกัน ให้ดูในตอน "นิทรรศการทางจอภาพ" ในไมโครฯ 92 ครับ)

- หากสร้างแผ่นภาพเปล่าไม่ได้ให้แจ้งให้ผู้ใช้ทราบ

- นำแผ่นภาพเปล่าไปปะติดกับดีซีเทียบเท่าของหน้าต่าง (ไม่มีไดรเวอร์) ด้วย SelectObject แล้วลอกภาพมาให้ด้วย BitBlt ซึ่งจะต้องถือว่าเป็นการลอกภาพจากหน้าต่างอีกเช่นกัน ถึงแม้พื้นที่ต้นทางจะอยู่นอกหน้าต่าง

ฟังก์ชัน BitBlt จะลอกภาพในแบบตรงตัวคือ ไม่ย่อหรือขยาย หากพื้นที่ปลายทางเล็กกว่าพื้นที่ต้นทางภาพที่เหลือ (พับจากมุมบนซ้ายเป็นหลัก) จะถูกตัดทิ้งไป ต้องการค่าพารามิเตอร์ 3 กลุ่มเช่นเดียวกับ StretchBlt แต่ในกลุ่มที่สองไม่ต้องบอกขนาดของพื้นที่ต้นทาง

การใช้ฟังก์ชันทั้งสองนี้คือ BitBlt กับ StretchBlt สามารถใช้แทนกันได้ตามความต้องการของโปรแกรม แต่ StretchBlt จะทำงานช้ากว่า

นำภาพออกแสดง

การนำภาพบิตแม็พที่เก็บไว้ใน mainBMP ออกแสดงหน้าต่างอยู่ในโปรแกรมของกิจกรรม BMPtoScrn ซึ่งเพนต์จะเป็นผู้เรียกใช้ ตามที่กล่าวในฉบับก่อน ที่นำมากล่าวเพื่อเป็นการเทียบเคียงกัน

- ต้องปะแผ่นภาพติดกับดีซีเทียบเท่าก่อนเช่นกัน

- ใช้ฟังก์ชันทั้งสองนี้เช่นกัน เพียงแต่กลับต้นทางกับปลายทาง คือให้ต้นทางเป็นแผ่นภาพ และปลายทางเป็นหน้าต่าง

การใช้ PatBlt

การใช้ฟังก์ชัน PatBlt จะนำมาใช้จริงในฉบับหน้า แต่ผมต้องการให้เรื่องของฟังก์ชันทั้งสามนี้อยู่ในตอนเดียวกัน ในฉบับนี้จึงเป็นการแนะนำตามรูปที่ 3 ซึ่งขออาศัยใช้รายการ

```
PROCEDURE  WndProC.cmCut(var msg : TMessage);
VAR  wndDC : HDC;  hOldObj : THandle;
     hbmp : HBitmap;  hbr,holdbr : HBrush;
CONST  zz : array[0..7] of Byte =
       ($FF, $F7, $EB, $DD, $BE, $7F, $FF, $FF);
BEGIN
  SwapPoint(sttpnt.x, endpnt.x);
  SwapPoint(sttpnt.y, endpnt.y);
  hbmp := CreateBitmap(8,8,1,1,@zz);
  hbr  := CreatePatternBrush(hbmp);
  UnrealizeObject(hbr);
  wndDC := GetDC(HWindow);
  hOldbr := SelectObject(wndDC, hbr);
  PatBlt(wndDC, sttpnt.x, sttpnt.y,
         endpnt.x-sttpnt.x, endpnt.y-sttpnt.y, PatCopy);
  SelectObject(wndDC, hOldbr);
  ReleaseDC(HWindow, wndDC);
  DeleteObject(hbr);   DeleteObject(hbmp);
END;
```

รูปที่ 3 แสดงการใช้ฟังก์ชัน PatBlt

Edit/Cut กล่าวคือ เมื่อรันโปรแกรมนี้แล้วให้ลอกหรือเก็บภาพใส่หน้าต่างตามที่กล่าวมา หมายพื้นที่ด้วย Edit/Mark แล้วเลือกรายการ Edit/Cut จะเกิดผลตามที่จะกล่าวต่อไป

คำว่า PatBlt หากอ่านตามตัว คือ Pattern block transfer แต่จะเป็นการดำเนินการตาม raster-operation กล่าวคือ PatBlt ต้องการพารามิเตอร์ 2 กลุ่ม กลุ่มแรกเป็นค่าดีซี ตำแหน่งและขนาดของพื้นที่ซึ่งคงต้องถือว่าเป็นพื้นที่ปลายทาง อีกกลุ่มมีค่าเดียวคือ ค่า rop ซึ่งจะให้ได้ดังนี้

- PatCopy คือตามที่ให้ไว้ในโปรแกรมในรูปที่ 3 จะเป็นการลอกแพตเทอร์นที่เราให้แก่แปรงคือ hbr เป็นแพตเทอร์นขาวดำขนาด 8×8 จุดภาพ มาให้พื้นที่ปลายทางด้วยการปู คือลอกซ้ำหลายครั้งจนเต็มพื้นที่

- PatInvert จะเป็นการ XOR แพตเทอร์นกับพื้นที่ปลายทาง

- PatPaint ให้เพนต์พื้นที่ปลายทาง

- DstInvert ให้กลับสี (เช่นดำเป็นขาว) ในพื้นที่ปลายทาง ด้วยวิธีการ XOR โดยไม่เกี่ยวกับแพตเทอร์น

- Blackness ทำให้พื้นที่ปลายทางเป็นสีดำทั้งหมด

- Whiteness ทำให้พื้นที่ปลายทางเป็นสีขาวทั้งหมด

ก็คงต้องทดลองทำความเข้าใจเพิ่มอีกเช่นกันครับ

การใช้งาน

เพื่อให้เข้าใจว่าทำไมผมถึงต้องการกิจกรรมที่จะกล่าวต่อไป จึงขอนำการใช้งานโปรแกรมนี้มากล่าวถึงครับ

ส่วนมากผมใช้จับภาพขนาดเล็กมาทำเป็นไอคอน จึงต้องนำมาขยายเพื่อให้เห็นถนัดจะได้ตัดได้ถูกต้อง จึงต้องการโปรแกรมสำหรับขยายและตัดภาพ ซึ่งเมื่อตัดแล้วจะส่งทางคลิปบอร์ดไปให้ไอคอนเอดิเตอร์เก็บลงไฟล์ในฐานะไอคอน หรือในบางครั้งการตัดภาพให้ได้ขนาด 32×32 จุดภาพก็ทำได้ยาก จึงต้องส่งผ่านคลิปบอร์ดไปใช้เพนต์บรัชขยายและตีกรอบก่อนเสร็จแล้วจึงนำมาตัด จึงต้องการโปรแกรมสำหรับติดต่อกับคลิปบอร์ด

หากคุณสงสัยว่าทำไมต้องยุ่งยากวิธีง่ายไม่มีหรือ มีครับ คือตามที่กล่าวในเรื่องไอคอนของขวัญ ในไมโครฯ 89 แต่มันง่ายไป

การขยายและย่อภาพ

ได้จัดรายการ Zoom In และ Zoom Out เพื่อการขยายและย่อภาพตามลำดับและได้จัดกิจกรรม cmZoomIn กับ cmZoomOut เพื่อสนองตอบเมสเสจจากรายการทั้งสองนี้ ดังโปรแกรมในลิสติงที่ 1.6 มีการทำงานคล้าย ๆ กัน คือ

- นำขนาดของปลายทางคือ ขนาดของรูปในหน้าต่างมาคูณหรือหารด้วย 2 เป็นการขยายหรือย่อ 2 เท่า

- ให้ cpyflg เป็นเท็จเพื่อเลือกใช้ StretchBlt ในการนำภาพบิตแม็พออกแสดง

- บังคับให้ทำงานที่เพนต์ด้วย InvalidateRect

- ให้ส่งเสียงดัง 1 ครั้งเพื่อสนองตอบการเลือกเมนูของผู้ใช้ เพราะการขยายภาพอาจใช้เวลาถึง 1 วินาที และจะส่งเสียงก่อนที่จะมีการขยาย เพราะฟังก์ชัน InvalidateRect จะทำให้เกิด

```
PROCEDURE  WndProC.cmZoomIn(var msg : TMessage);
BEGIN
  dstwid := dstwid * 2;  dsthi := dsthi * 2;
  cpyflg := false;
  InvalidateRect(HWindow,nil,true);
  MessageBeep(100);
END;


PROCEDURE  WndProC.cmZoomOut(var msg : TMessage);
BEGIN
  dstwid := dstwid div 2;  dsthi := dsthi div 2;
  cpyflg := false;
  InvalidateRect(HWindow,nil,true);
  MessageBeep(100);
END;
```

ลิสติงที่ 1.6

การส่งเมสเสจ wmPaint ซึ่งการถอนเมสเสจจะเกิดขึ้นต่อเมื่อจบการทำงานจากโปรแกรมส่วนนี้แล้ว

การตัดภาพ

การตัดภาพจะดำเนินการได้โดยกำหนดรอยตัดด้วยการตีกรอบหมายพื้นที่ แล้วเลือกวิธีตัดภาพซึ่งมีอยู่ 2 วิธีตามรายการในเมนู คือ

- File/Get/Mark กิจกรรม cmGetMark เป็นการตัดภาพตามรอยตัดและขนาดของภาพที่ปรากฏในหน้าต่างขณะนั้น หากเป็นภาพขยายก็จะได้เป็นภาพขยายด้วย

- File/Get/BMP Mark กิจกรรม cmGetMarkBMP เป็นการตัดภาพโดยอาศัยรอยตัดจากภาพในหน้าต่างซึ่งควรเป็นภาพขยาย ส่วนการตัดภาพจริงจะตัดที่บิตแม็พซึ่งไม่ได้ขยาย จึงได้ภาพขนาดปกติ

โปรแกรมของกิจกรรมทั้งสองนี้อยู่ในลิสติงที่ 1.7 โดยที่

cmGetMark เป็นการเก็บภาพในหน้าต่างตามปกติและจะเก็บตรงรอยตัด จึงต้องลบเส้นปะออกก่อน ด้วย DoMarkOff แล้วจึงลอกภาพให้แก่ mainBMP ด้วย ScrnToBMP

```
PROCEDURE  WndProC.cmGetMark(var msg : TMessage);
BEGIN
  DoMarkOff;  cpyflg := true;
  ScrnToBmp(mainBMP, mainwid,mainhi);
  SetDefault;
  InvalidateRect(HWindow,nil,true);
END;


PROCEDURE  WndProC.cmGetMarkBMP(var msg : TMessage);
VAR   ztme,sx1,sy1,sx2,sy2 : Integer;
BEGIN
  if  dstwid < mainwid  then  Exit;
  ztme := dstwid div mainwid;
  sx1 := (sttpnt.x - dstx) div ztme;
  sy1 := (sttpnt.y - dsty) div ztme;
  sx2 := (endpnt.x - dstx) div ztme;
  sy2 := (endpnt.y - dsty) div ztme;
  mainwid := sx2 - sx1;  mainhi := sy2 - sy1;
  if  scndBMP <> 0  then  DeleteObject(scndBMP);
  scndBMP := mainBMP;  mainBMP := 0;
  BMPtoBMP(mainBMP,0,0,scndBMP,sx1,sy1,mainwid,mainhi);
  SetDefault;
  InvalidateRect(HWindow,nil,true);
END;
```

ลิสติงที่ 1.7

```
PROCEDURE  WndProC.BMPtoBMP(var dstnBMP : HBitmap; dx,dy : Integer;
                            srceBMP : HBitmap; sx,sy,swid,shi : Integer);
VAR   oldobj1,oldobj2 : THandle;  wndDC,dstnDC,srceDC : HDC;
BEGIN
  wndDC := GetDC(HWindow);
  if  dstnBMP = 0  then
    dstnBMP := CreateCompatibleBitmap(wndDC, swid,shi);
  dstnDC := CreateCompatibleDC(wndDC);
  srceDC := CreateCompatibleDC(wndDC);
  ReleaseDC(HWindow, wndDC);
  oldobj1 := SelectObject(dstnDC, dstnBMP);
  oldobj2 := SelectObject(srceDC, srceBMP);
  BitBlt(dstnDC, dx,dy,swid,shi, srceDC, sx,sy, srcCopy);
  SelectObject(dstnDC, oldobj1);
  SelectObject(srceDC, oldobj2);
  DeleteDC(dstnDC);  DeleteDC(srceDC);
END;
```

ลิสติงที่ 1.8

```
FUNCTION   GetFiNameDlg(awnd : HWnd; filter, capt,filespec : PChar;
                        openNOTsave : Boolean) : Boolean;
VAR  ofn : TOpenFileName;
     finame,path,filtr : array[0..255] of Char;
     usrflt : array[0..40] of Char;
     id,ipair : Word; Pext,Pusr : PChar;  res : Boolean;  ch : Char;
BEGIN
  StrCopy(path, filespec);  Pext := StrRScan(path, '\');
  if  Pext <> nil  then
    begin  Pext[0] := #0;  Pext := Pext + 1  end
  else  Pext := @path;
  StrCopy(finame, Pext);    Pext := StrRScan(finame, '.') + 1;

  StrCopy(filtr, filter);  CheckFilter(filtr, Pext, ipair);
```

รูปที่ 4 ส่วนของโปรแกรมที่ให้แก้โปรแกรมของฟังก์ชัน GetFiName เดิม

cmGetMarkBMP ได้ให้ดำเนินการดังนี้

- คำนวณรอยตัดใหม่ โดยคิดจากอัตราขยาย ztme
- ในการตัดภาพได้ย้ายบิตแม็พไปไว้กับ ScndBMP เพื่อจะให้ตัดคือลอกส่วนที่ต้องการมาให้แก่ mainBMP ใหม่ด้วย BMPtoBMP ซึ่งมีโปรแกรมอยู่ในลิสติงที่ 1.8 ขอให้ดูเทียบกับลิสติงที่ 1.5 ครับ

โปรแกรมท่อนต่อไป

ถึงฉบับนี้คุณสามารถใช้โปรแกรม ScreenCaptor ในการจับภาพ ลอกภาพ ตัดภาพ เก็บลงไฟล์ อ่านกลับมารวมทั้งอ่านไฟล์ .BMP อื่นด้วย เพื่อนำออกแสดง และการพิมพ์ภาพ

คงขาดการติดต่อกับคลิปบอร์ดอันเป็นเรื่องสำคัญเรื่องหนึ่งในการเขียนโปรแกรมบนวินโดวส์ ซึ่งความจริงแล้วไม่ใช่เรื่องยาว แต่ผมต้องการแยกตอนจึงขอนำไปไว้ในฉบับหน้าครับ

ขอแก้โปรแกรมเก่า

ในไมโครฯ 94 ได้สร้างฟังก์ชัน GetFiName ไว้ เพื่อใช้ฟังก์ชัน GetOpenFileName หรือ GetSaveFileName อีกทอดหนึ่งนั้น และได้ตั้งชื่อเลียนชื่อสองฟังก์ชันนี้ แต่ต่อมารู้สึกงงทุกครั้งที่นำมาใช้ครับ จึงขอแก้ชื่อเพื่อบ่งบอกว่าเป็นไดอะล็อกบ็อกซ์คือ GetFiNameDlg และขอแก้โปรแกรมตอนต้นให้เป็นดังรูปที่ 4 และขอให้แก้โปรแกรมที่เรียกใช้ฟังก์ชันนี้ด้วยครับ

หนังสือดี ซีเอ็ด

รู้กฎหมายอยู่กับตัว ไม่ต้องกลัวเป็นความ

■ โดย เถกิงศักดิ์ คำสุระ
ราคาเล่มละ 165 บาท

เป็นหนังสือที่รวบรวมความรู้ทางกฎหมายที่น่าสนใจ มีความสัมพันธ์กับชีวิตประจำวัน และในการประกอบอาชีพของนักธุรกิจ ประชาชนทั่วไป เนื้อหาภายในเล่มประกอบด้วยความรู้ทางกฎหมายเรื่อง บุคคลและทรัพย์สิน ครอบครัว การหมั้น การสมรส มรดก เรื่องน่ารู้เกี่ยวกับคดีล้มละลาย สิทธิของลูกจ้าง เมื่อถูกเลิกจ้าง อาคารชุด ทรัพย์สินทางปัญญา ฯลฯ ตลอดจนกฎหมายเกี่ยวกับวิธีพิจารณาความอาญาและคดีแพ่ง ตั้งแต่เริ่มต้นจนเสร็จสิ้นกระบวนความ

หาซื้อได้ตามร้านหนังสือทั่วไป
หรือสั่งซื้อโดยตรงได้ที่

บริษัท ซีเอ็ดยูเคชั่น จำกัด
800/43-45 ซอยตระกูลสุข ถนนอโศก-ดินแดง ห้วยขวาง กรุงเทพฯ 10400 โทร. 248-6280-9 (เวลาทำการ จันทร์-ศุกร์ 8.10-17.30 น.) โดยส่งธนาณัติสั่งจ่าย ปท.ดินแดง ในนามบริษัทฯ หรือเช็คส่วนตัวของธนาคารในเขตกรุงเทพฯ พร้อมค่าจัดส่ง

ค่าจัดส่ง สั่งซื้อไม่เกิน 100 บาท ค่าจัดส่ง 8 บาท สั่งซื้อเกิน 100 บาท ค่าจัดส่ง 8%

พิเศษ สำหรับสมาชิกวารสารของซีเอ็ด สั่งซื้อโดยตรงที่บริษัทฯ ได้ในราคาลด 15% เพิ่มค่าลงทะเบียน 4 บาท ค่าจัดส่งฟรี โดยระบุหมายเลขสมาชิกด้วย